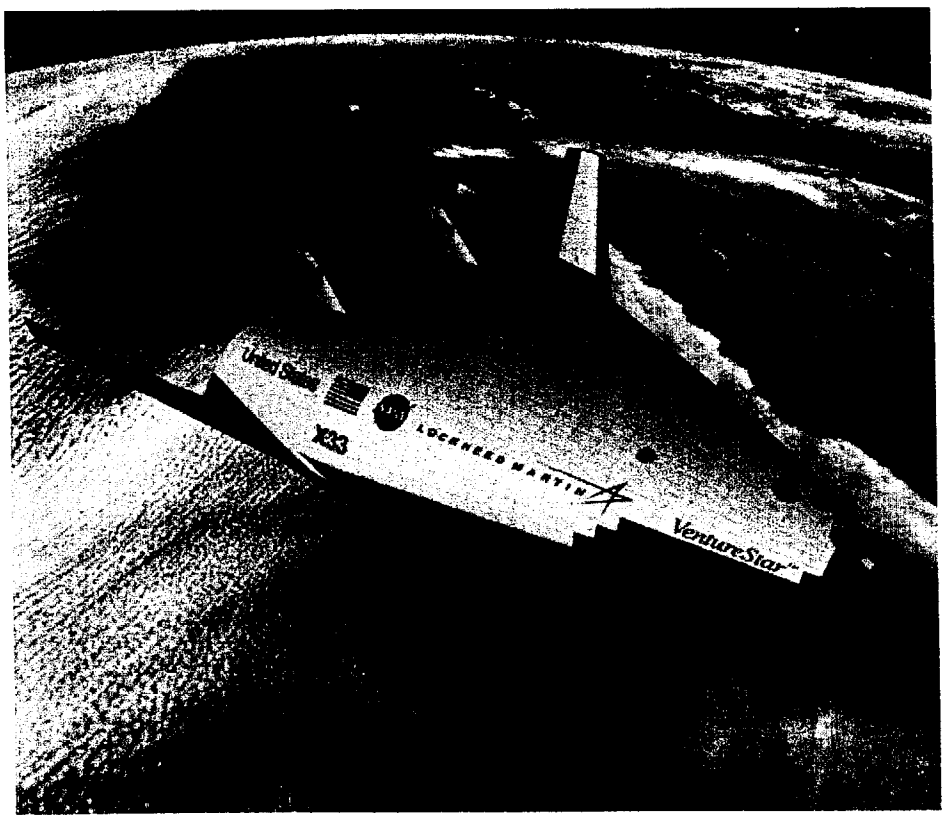
Information Sciences & Technology
Ames Research Center

NASA



Tera Scale Systems and Applications

Chuck Niggley
User Services Director
NASA Advanced Supercomputing Division
NASA Ames Research Center
Moffett Field, CA 94035-1000

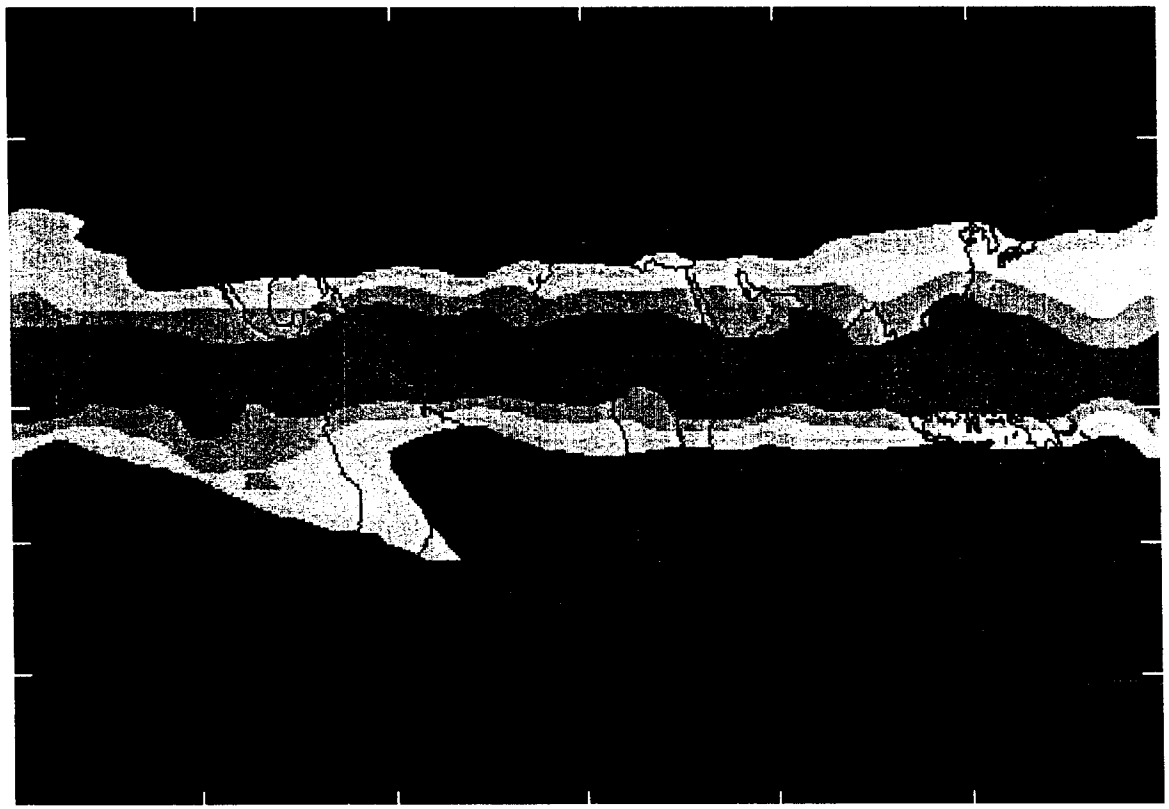
NASA's Computational Challenges

- Aeronautics
  - X-Vehicle Development
  - Shuttle Upgrades
  - Airspace Control and Simulation
- Astrobiology
  - Protein Folding and Beyond
- Earth Science
  - Atmospheric Physics
  - Oceanography
- Nano Technology
  - Chemistry
- Space Science
  - Stellar Dynamics
  - Chemistry/Spectra
  - Instrument Data Analysis and Support
  - Planetary Modeling

NASA Advanced Supercomputing Facility

Current SGI Inventory -- 20+ systems

# CPUs	System	Name
1024	O3K	chapman
512	O3K	lomax3
128	O2K	steger
64	O2K	hopper
32	O2K	turing
32	O2K	fermi
128	O2K	kalnay
128	O2K	jimpf1
64	O2K	sunrise
16	O2K	lou
16	O2K	evelyn
16	O2K	piglet
2/2	O2K	this/that



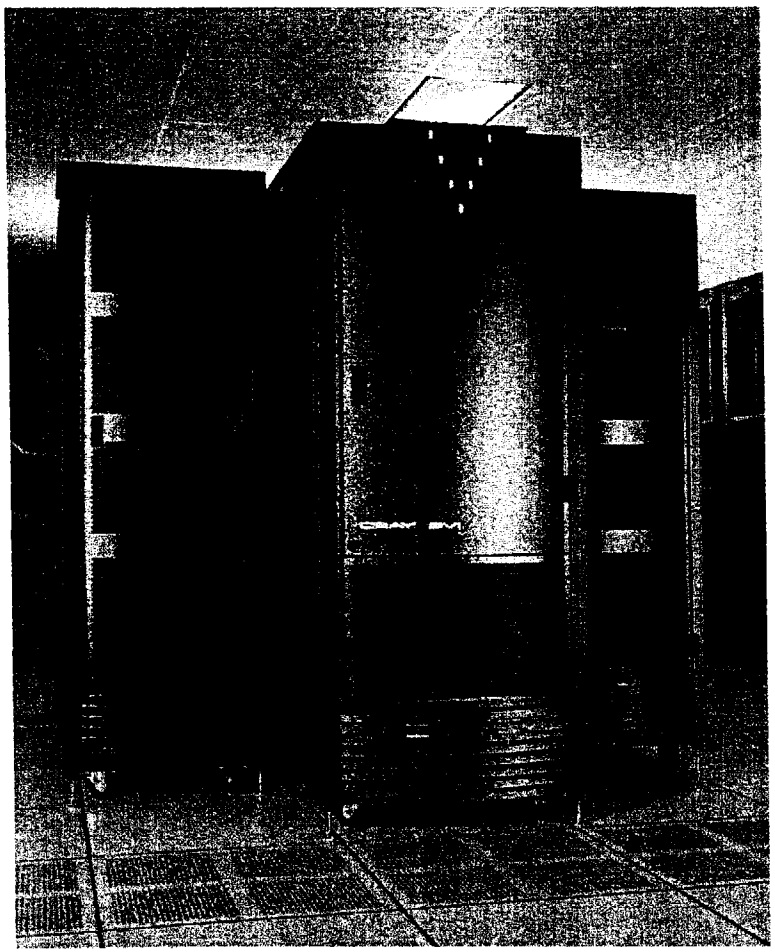
Goal: Production Quality Highly Parallel Supercomputer

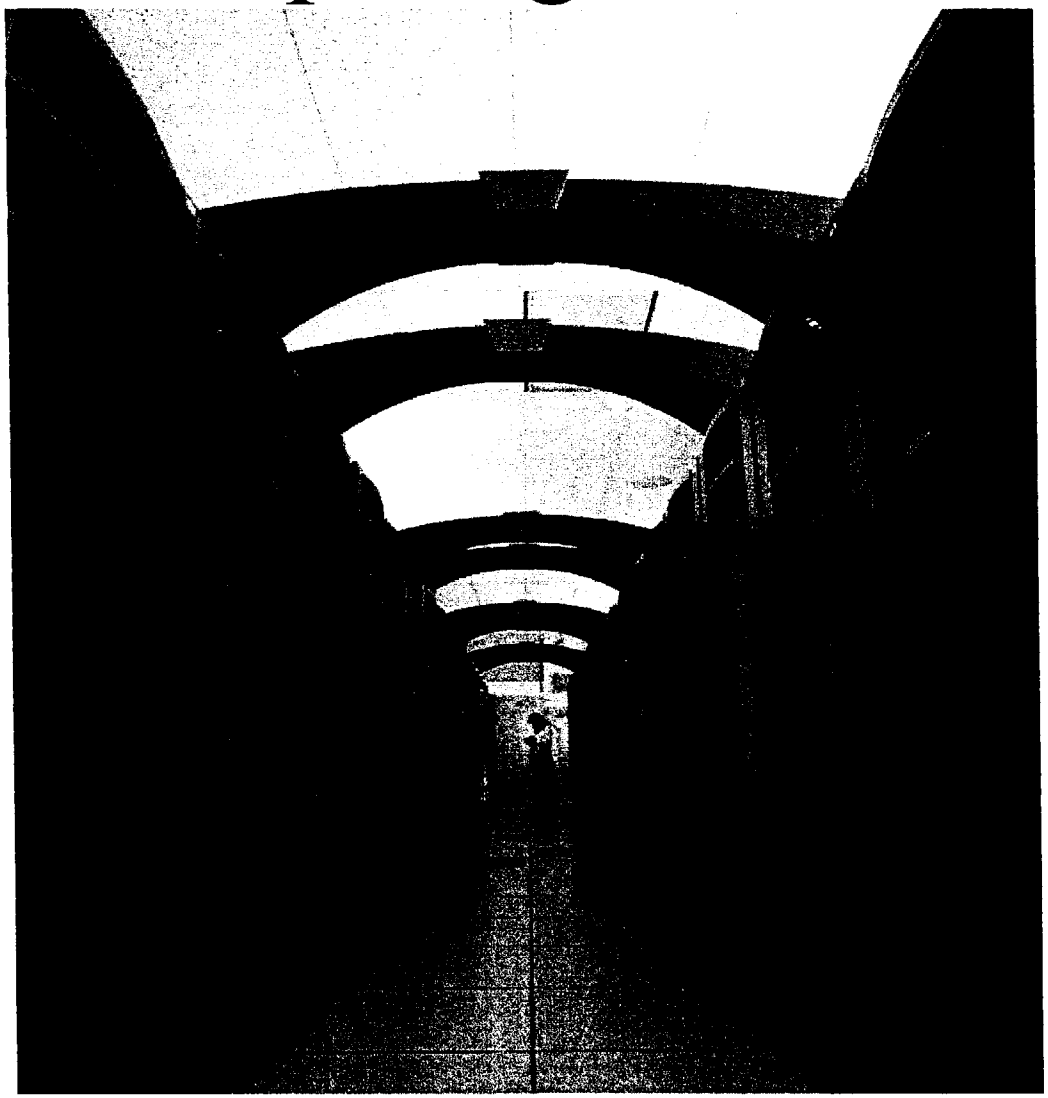
- What is a Production Supercomputer?
  - Significantly faster that previous generation
  - Less Expensive
  - Productive Environment for the Users (easy to use)
  - As reliable as the "Gold Standard"
- Have to move into the realm of hundreds or thousands of processors
  - Its all about the interconnect
- Many Attempts (these all failed…)
  - Connection Machine - CM2 (performance)
  - Connection Machine - CM5 (performance)
  - Intel - IPSC-860 (performance)
  - Intel – Paragon (performance)
  - IBM - SP2 (performance)
  - SGI - Power Challenge Cluster (reliability)
  - Cray - J90 Cluster (vendor backed out of commitment)

Information Sciences & Technology
Ames Research Center

High-End Computing

NASA



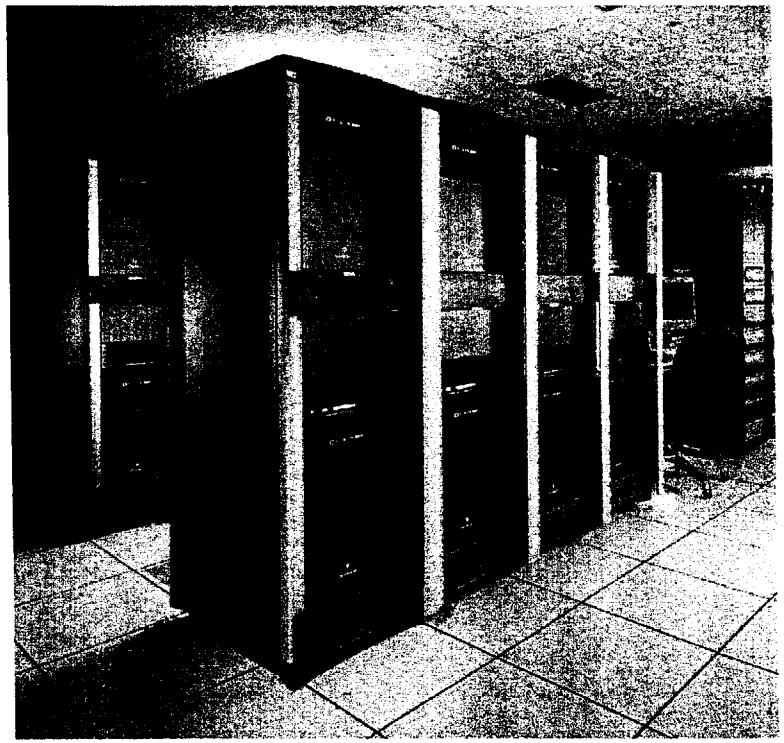


Barriers to Scientists Obtaining a SuperComputing Capability for Their Research

- There are many problems that require supercomputing
- However, it's typical that applications may require substantial modifications to achieve even moderate levels of parallelism.
- This can translate into several man years of effort, and when you have done this, you'll likely not run faster than a C90 supercomputer manufactured in 1993.
- Many (most) productive scientists are simply unable to access supercomputing because it is either difficult or even not possible to effectively scale their applications.
- For example,
  - Computational Chemistry
  - DAO
  - Molecular Dynamics
  - Space Science

Shared memory can go a long way towards SIMPLIFICATION.

High-End Computing



QuickTime™ and a
Cinepak decompressor
are needed to see this picture.

SGI O3K 1024 CPU System

- 2 operational 512p 400mhz O3000 systems
  - testing and fallout
  - staff and small group of users on systems
- 1024p in initial topology configuration in Sep
- Moved to higher bandwidth/lower latency topology in October
- Processor speed upgrade (600 Mhz) April 2002
- Expect to sustain 20% of peak - 200 Gflops

SGI O3K 1024 CPU System

What Happened this past year:

- Hardware
  - Memory failures -- 10
  - CPU failures -- 15
  - Router failures -- 6
  - Unkown/Other -- 28
- Software
  - Kernel changes -- 15
  - MPI/User Code Hangs -- 20

PBS MOM Changes

- pbs_mom:

(1) new configuration options: 'schd_clump_size' = number of nodes in a "clump" (should be the same as the scheduler's 'NODES_PER_CLUMP')

'collector_interval' = number of seconds allowed for the collector thread to finish one scan

(2) new PBS environment variable 'PBS_CPUS' which says just which CPUs a job is using (in a compact representation)

(3) setting 'OMP_NUM_THREADS', 'MP_SET_NUMTHREADS', 'OMP_DYNAMIC', 'MPC_GANG', 'MPI_UNBUFFERED_STDIO'

PBS Scheduler Enhancements

(1) new charging algorithm

(2) software-defined "clumps" rather than previous "hardwired" hardware "clumps" of 64 nodes

(3) new configuration options:

'LOCAL_DEDTIME_FILE' - allows easy implementation of "emergency" dedtimes

'SCHEDULE_TZ' - basically, setting of 'TZ' needed for 'schedule' output

'NODES_PER_CLUMP' - size in nodes of software-defined "clumps"

'STUCK_TOO_LONG' - number of seconds to wait before dumping out process information about a stuck cpuset

'ALLOW_CPUSET' - list of usernames allowed to use 'qsub -l cpuset='

PBS Scheduler Enhancements

(4) handling of 'qsub -l cpuset=<nodemask>' which lets (empowered) users specify the cpuset a job should be run in

(5) confining jobs to nodes specified in queue's assigned nodemask (allowing the system to be partitioned, if that's thought necessary)

(6) on first iteration after server comes up, try to rerun any previously-running jobs in their original cpuset

Where Are We Today

Not quite there yet

- BUT, since Irix 6.5.16 installed system up 10 days
- POP Cross-Interference when run against other codes (Parallel Ocean Modeling)
- Slow MPI startup due to cross mapping of address space

LOTS of progress HAS been made due to SGI and NAS staff working together to resolve issues

Where to go from here

Applications

- More instrumentation to measure effects of bandwidth on performance.
- More instrumentation to precisely measure memory layouts and effects on performance.
- Work on general code scaling

Systems Software

- Improve some existing scaling issues in IRIX
- Scaling issues exist in VM and I/O subsystems

Performance - The Focus is on Parallelism

Parallelism is the key to performance on any system manufactured today. If you don't scale to hundreds of CPUs, you won't get to the 100+ GFLOPS you need today to stay competitive in high end computing. Parallelism was being aggressively pursued on two fronts. Now there are three.

- *Message Passing Interface (MPI)*
  - *Arcane and complex user interface - 100 routines, 50,000+ lines of source*
  - *Explicit "messages" – large latencies – very slow*
  - *User provides all parallel decomposition/code modification*
  - *Often requires simplification of physics for scaling*

- *Shared Memory Parallelism (OpenMP)*
  - *Really acceptable only for small processor counts*
  - *Very difficult to scale to 100's of CPU's without major rewrite*

- *NASA's Shared Memory Multi-Level Parallelism (MLP)*
  - *Simple extension to Cray parallel/vector programming model - 3 routines, 150 lines of source*
  - *No messaging - All communication via shared memory*
  - *Much easier to build/port code than MPI (Man months vs. Man years)*
  - *Minimum changes OVERFLOW (MPI/MLP=20,000/800 lines), FVCORE (8000/400 lines)*
  - *Dramatically better performance with increasing processor count*

What is MLP?

Shared Memory Multi-level Parallelism (MLP) is the utilization of multiple levels of parallelism within an application executing on a NUMA based system architecture in order to increase its parallel efficiency during execution. It is an open system design (runs on any SMP) and has the following attributes:

- ***Two levels of parallelism (the so-called "hybrid" approach)***
- ***Coarse grained parallelism provided by Unix forked processes***
- ***Fine grained parallelism provided by the compiler at loop level (OpenMP)***
- ***No messaging - communication through "global" common blocks***
- ***Targeted for the new large CPU count NUMA SMP systems***
- ***But method has been adapted to execute across small clusters as well***

MLP - A New Concept for Multi-zonal CFD

NASA's Multi-zonal CFD codes like OVERFLOW, CFL3D, LAURA, INS3D, and TLNS3D to name a few, are ideal candidates for MLP parallelism. These codes decompose a large region of interest into many linked smaller 3D regions. These smaller regions can be solved mostly in parallel, with the occasional exchange of boundary information at the end of a time step.

In short, the recipe for converting a multi-zonal CFD code to MLP is:

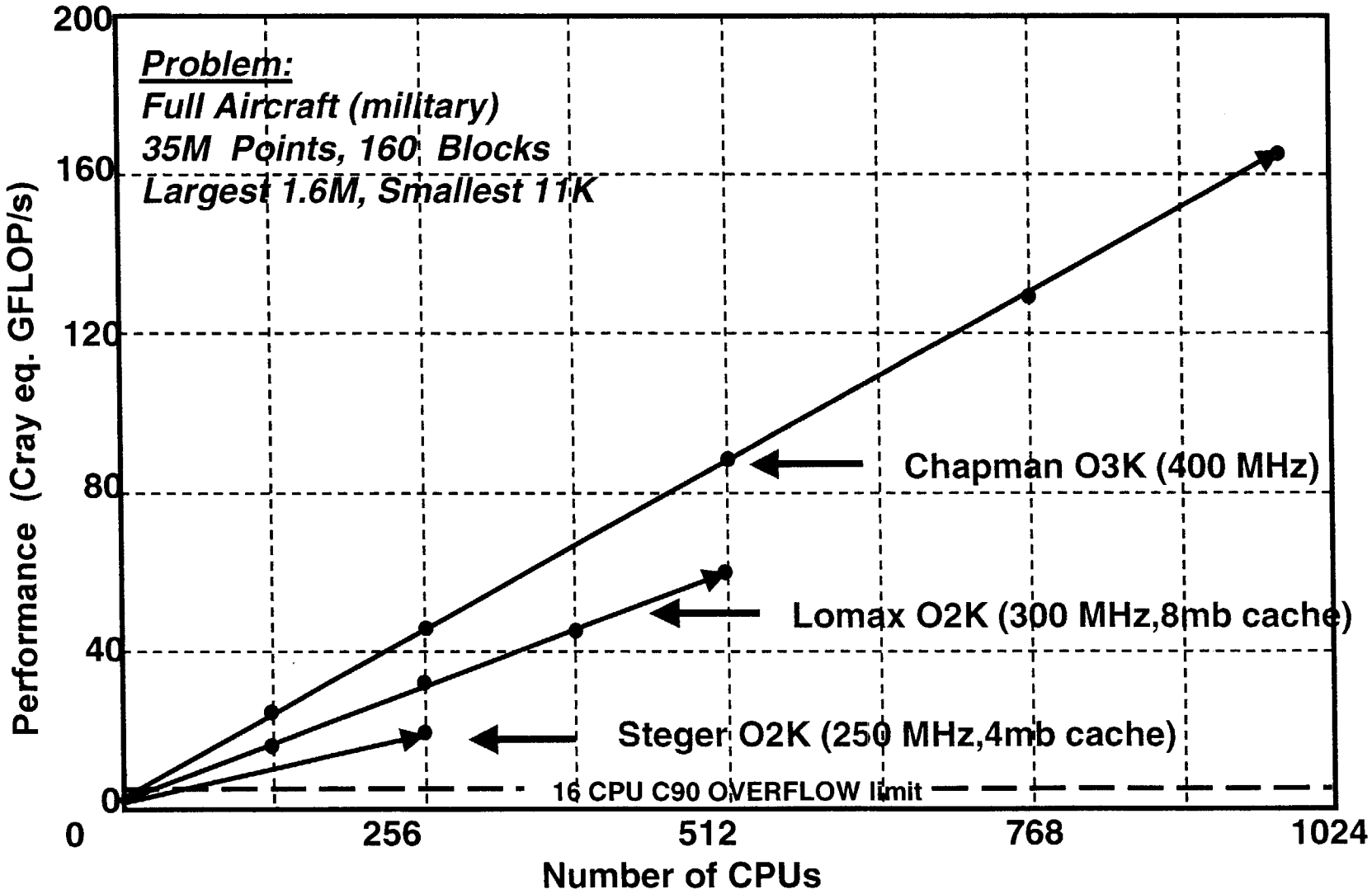
- *Spawn MLP parallel processes*
- *Assign groups of 3D zones to each MLP process*
- *Solve the groups of zones in parallel*
- *Assign groups of CPUs to each MLP process*
- *Use the CPUs in a group for fine grained parallelism for each zone*
- *Use shared memory arenas to hold all global data (BCs etc)*
- *Synchronize computation as needed with barriers*

Success Stories

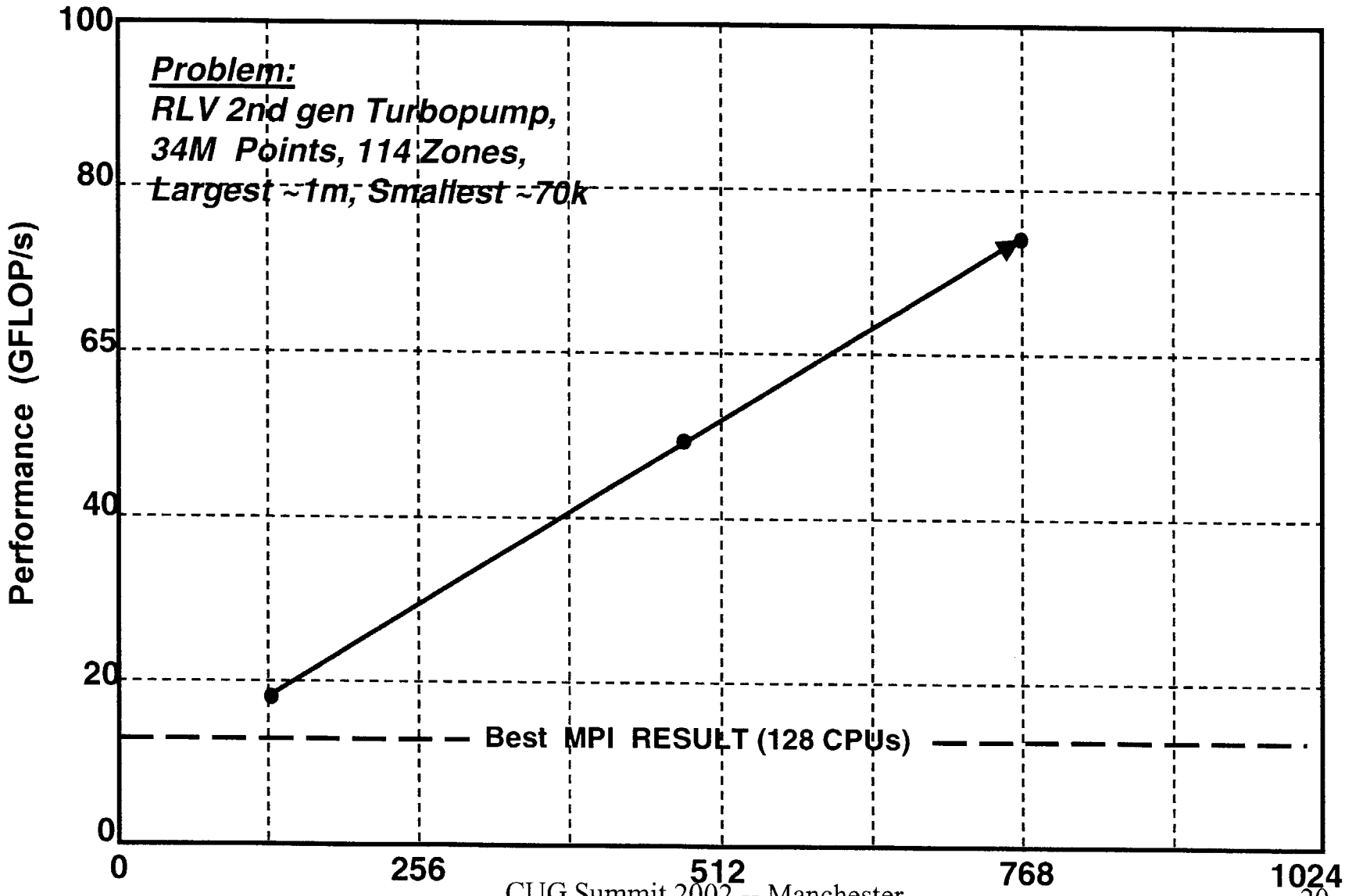
- OVERFLOW-MLP
- INS3D
- CACTUS

OVERFLOW-MLP Performance vs CPU Count

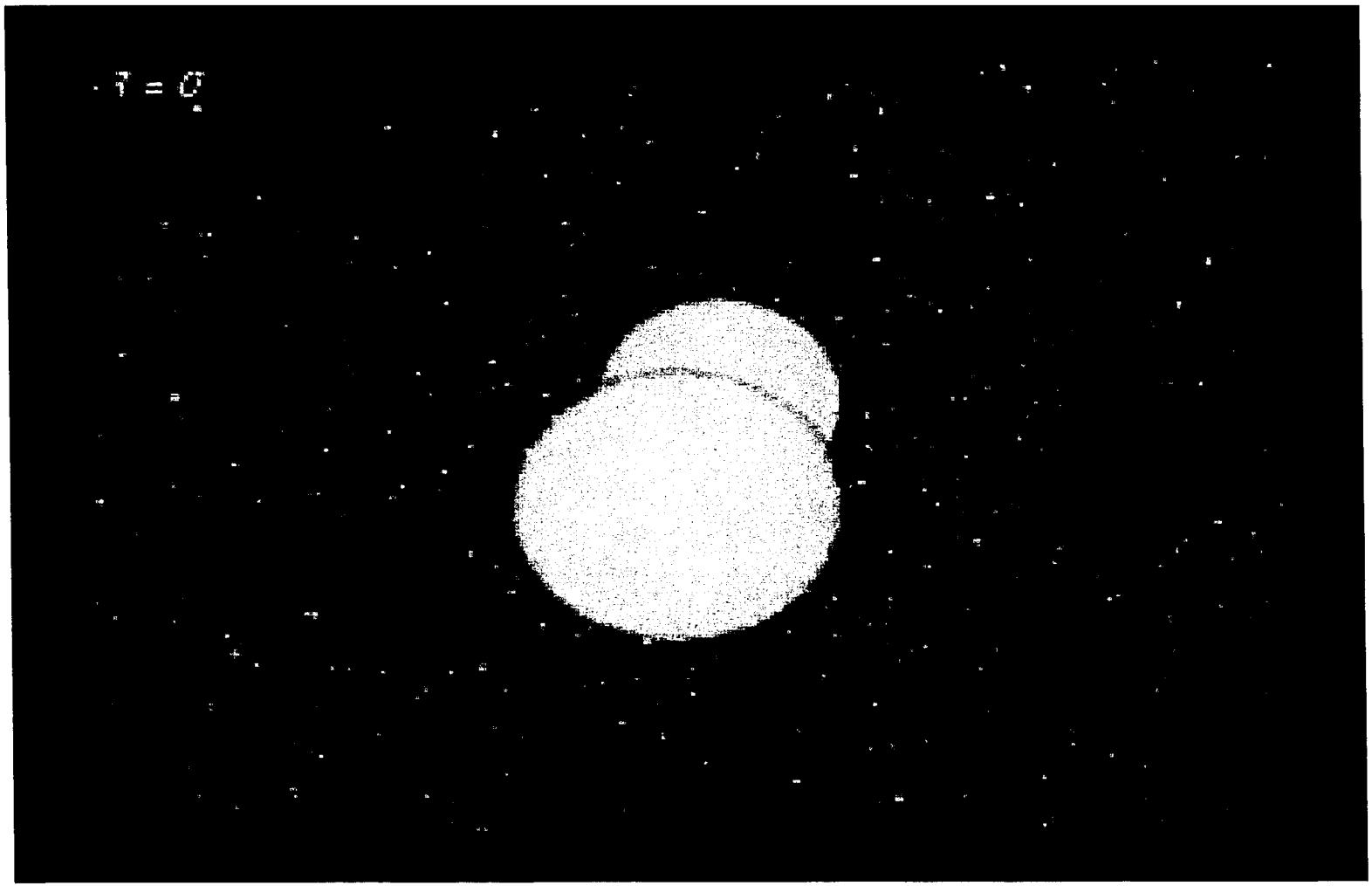
Systems: 1024 CPU O3K , 256&512 CPU O2KS



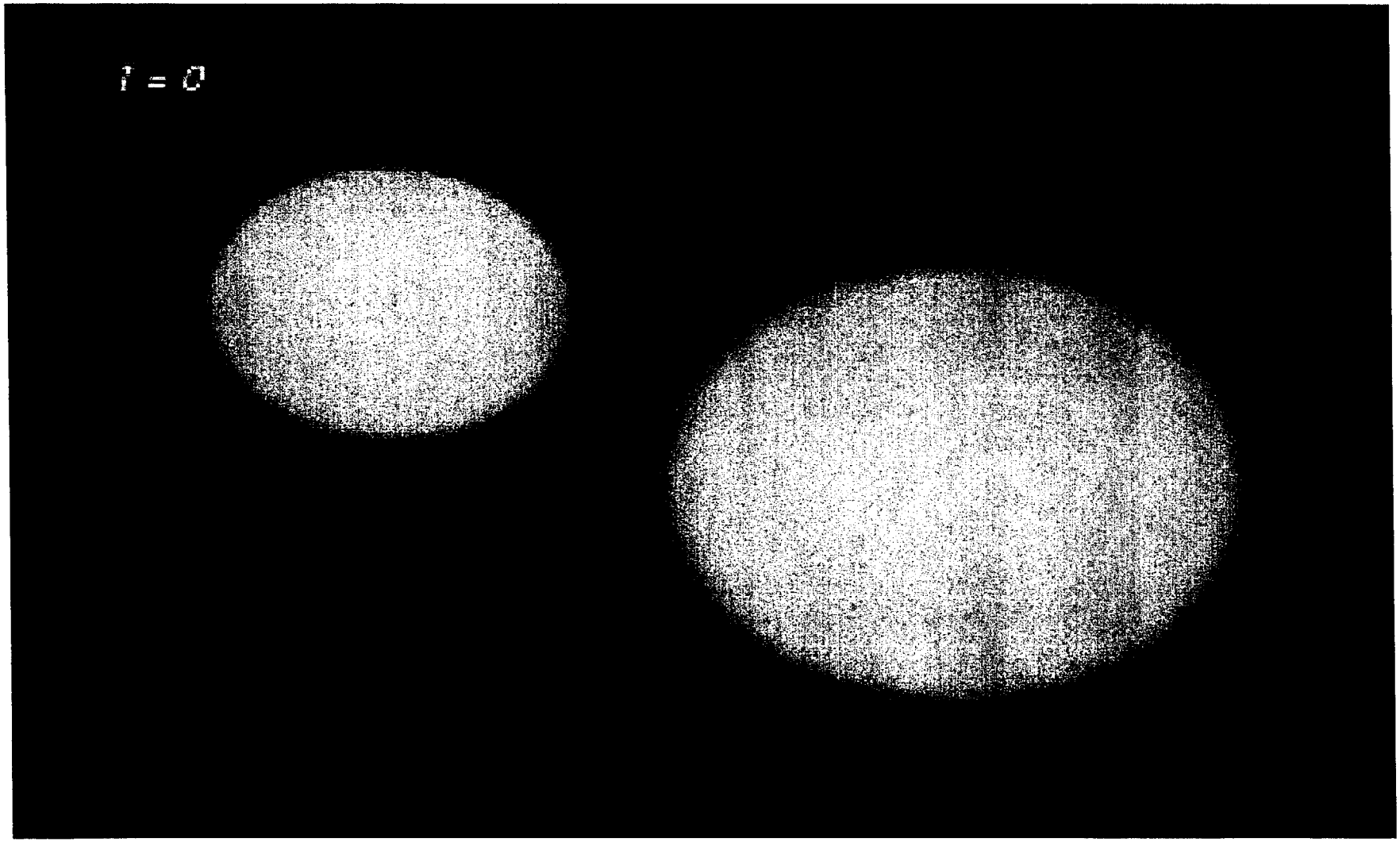
INS3D-MLP Performance vs CPU Count
System: 1024 CPU O3K (400 MHz)



Newtonian inspiral of two neutron stars



Head on collision of two neutron stars



Computational Nanosciences

QuickTime™ and a
Cinepak decompressor
are needed to see this picture.

QuickTime™ and a
Cinepak decompressor
are needed to see this picture.

Further Contact

- http://www.nas.nasa.gov
- support@nas.nasa.gov
- 1 800 331 USER (8737)
- 650 604 4444
- Ciotti@nas.nasa.gov 650 604 4408

Special Thanks

- Bob Ciotti
- John Pandori
- Francois Montoya
- Ryan Coburn
- Gabe Wedekind
- Alan Powers
- Chris Hamilton
- Ekechi Nwokah